UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/777,869	02/12/2004	Dennis Steven DeLorme	ROC920040004US1	6102

30206 7590 06/13/2008
IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
KIM, PAUL	

ART UNIT	PAPER NUMBER
2161	

MAIL DATE	DELIVERY MODE
06/13/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/777,869
Filing Date: February 12, 2004
Appellant(s): DELORME ET AL.

Scott A. Stinebruner
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1 April 2008 appealing from the Office action mailed 1 November 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2003/0037302	Dzienis	06-2002
5,625,804	Cooper	04-1995
5,873,097	Harris et al	12-1996
6,571,231	Sedlar	05-2002
2003/0217057	Kuroiwa et al	05-2003
6,728,907	Wang et al	04-2000
6,338,072	Durand et al	07-1998

West et al, "Sams Teach Yourself Macromedia Fireworks MX in 24 Hours," published on 4 December 2002.

Dubinski, J., "Non-recursive tree walks," published on 1 May 1996.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. **Claims 1, 2, 7-9, 20-22, 24-26 and 28** are rejected under 35 U.S.C. 103(a) as being unpatentable over Dzienis (USPGPUB No. 2003/0037302, hereinafter referred to as DZIENIS), filed on 21 June 2002, and published on 20 February 2003, in view of Cooper (U.S. Patent No. 5,625,804, hereinafter referred to as COOPER), filed on 17 April 1995, and issued on 29 April 1997.

3. **As per independent claims 1 and 24, and dependent claim 22**, SEDLAR, in combination with COOPER, discloses:

> A method for converting a filesystem from a first type to a second type, the method comprising the steps of:
>
> generating a list of directories of the first type in the filesystem to convert {See DZIENIS, [0040], wherein this reads over "an inventory is performed by scanning the directory containing files to be converted"}; and

converting each directory in the list to the second type {See DZIENIS, [0042], wherein this reads over "[t]he file conversion is then performed on each file for every file that is supported"} while maintaining the file system in an active state {See COOPER, C2:L44-59, wherein this reads over "[t]he data conversion technique of the present invention advantageously enables data to be converted from one format to another while maintaining system operations"}.

The combination of the inventions disclosed in DZIENIS and COOPER would disclose a method wherein a list of directories to convert is generated (i.e. "the directory containing files to be converted"), and the directories therein are converted to a second type (i.e. is converted to another format). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS and COOPER.

One of ordinary skill in the art would have been motivated to do this modification so that the directory structure of filesystems may be updated accordingly or converted to another filesystem type.

4. **As per dependent claims 2 and 21**, DZIENIS, in combination with COOPER, discloses:

The method of claim 1, further comprising the step of:

> sequentially initiating the steps of generating and converting upon initial program load of a computer system utilizing the filesystem {See COOPER, C1:L28-40, wherein this reads over "new versions of the software are loaded into the system along with change instructions providing information controlling the update"; and C7:L34-36, wherein this reads over "conversion begins for the rest of the data records to be converted. Initially, a variable N is initialized to the next record to be converted"}.

The combination of the inventions disclosed in DZIENIS and COOPER would disclose a method wherein the generation and conversion of the list of directories is initiated upon an initial program load of the computer system. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS and COOPER.

One of ordinary skill in the art would have been motivated to do this modification in order to being the processes of generation and conversion.

5. **As per dependent claim 8**, DZIENIS, in combination with COOPER, discloses:

The method of claim 1, further comprising the step of:

> marking a particular directory as being in the process of conversion once the particular directory is in the list {See COOPER, C6:L10-15, wherein this reads over "a further determination is made as to whether data record 1 has a status of converting . . . That is, are the data records specified by record 1 in the process of being converted?"}.

The combination of the inventions disclosed in DZIENIS and COOPER would disclose a method wherein the particular directory is marked as being in the process of conversion once the particular directory is in the list (i.e. retains the status of converting). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS and COOPER.

One of ordinary skill in the art would have been motivated to do this modification so that the user and the conversion process may be aware of the active process of conversion.

6. **As per independent claims 20, 25 and 26**, DZIENIS, in combination with COOPER, discloses:

A method for converting a filesystem from a first type to a second type, the method comprising the steps of:

> executing a conversion process to convert each directory of the first type in the filesystem into the second type while maintaining the filesystem in an active state {See COOPER, C2:L44-59, wherein this reads over "[t]he data conversion technique of the present invention advantageously enables data to be converted from one format to another while maintaining system operations"}; and
>
> terminating the conversion process when every directory of the first type in the filesystem has been converted to the second type.

The termination of the conversion process would be inherent to the claimed invention since it is necessary that the conversion process terminate once it reaches the last directory in the list to convert.

7. **Claims 3-5, 9, 23, 28 and 34** are rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Official Notice.

8. **As per dependent claims 3, 4 and 23**, DZIENIS, in combination with COOPER and Official Notice, discloses:

The method of claim 1, wherein the step of converting further includes the steps of:

> retrieving an identifier of a directory in the list {See DZIENIS, [0040], wherein this reads over "scanning of the directory containing files to be converted"};
>
> converting the directory to a second-type directory {See COOPER, C5:L18-20, wherein this reads over "[a]fter all of the complexes receive the new code, the data records are converted from the old format to the new format and stored"}; and
>
> activating the second-type directory.

The Examiner takes Official Notice that It would have been obvious to one of ordinary skill in the art that at the time a directory is converted to a second-type directory, prompting the creating of a second-type directory, that the second-type directory would be activated and available for access.

9. **As per dependent claim 5**, the Examiner takes Official Notice that it would have been obvious and widely-known to one of ordinary skill n the art to have a list represent a top-down view of the filesystem spanning from a root directory down to an outermost leaf-node.

10. **As per dependent claims 9 and 28**, the Examiner takes Official Notice that it would have been obvious and widely-known to one of ordinary skill n the art to append a new object to the end of a directory.

11. **As per dependent claim 34**, DZIENIS, in combination with COOPER and Official Notice, discloses:

> The apparatus of claim 26, wherein the program code is further configured to: for a particular directory already converted, convert the particular directory back to the first type {See COOPER, C5:L18-20, wherein this reads over "[a]fter all of the complexes receive the new code, the data records are converted from the old format to the new format and stored"}.

The Examiner takes Official Notice that It would have been obvious to one of ordinary skill in the art that a directory which has been converted may be reverted back to its original type using the processes of conversion again.

12. **Claim 6** is rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Harris et al (U.S. Patent No. 5,873,097, hereinafter referred to as HARRIS), filed on 17 December 1996, and issued on 16 February 1999.

13. **As per dependent claim 6,** DZIENIS, in combination with COOPER and HARRIS, discloses:

> The method according to claim 5, wherein the step of converting each directory is performed for each directory in an order opposite to that of the list {See HARRIS, C7:L2-8, wherein this reads over "[t]he procedure then works the way back up the chain, performing the changes on the in-memory structure"}.

The combination of the inventions disclosed in DZIENIS, COOPER, and HARRIS would disclose a method wherein the step of converting each directory is performed for each directory in an order opposite to that of the list (i.e. the process works its way back up the chain). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and HARRIS.

One of ordinary skill in the art would have been motivated to do this modification since this reverse order would permit the detection of old-style directories that may have been introduced after the list of old-style directories was built.

14. **Claim 7** is rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Sedlar (U.S. Patent No. 6,571,231, hereinafter referred to as SEDLAR), filed on 28 May 2002, and issued on 27 May 2003.

15. **As per dependent claim 7**, DZIENIS, in combination with COOPER AND SEDLAR, discloses:

> The method according to claim 1, wherein the step of generating further includes the steps of:

a) adding a root directory as a current entry in the list {See SEDLAR, Figures 1-3; and C3:L50-61, wherein this reads over "directory links table"};

b) identifying a child directory of the current entry in the list {See SEDLAR, Figures 1-3; and C3:L50-61, wherein this reads over "directory links table"};

c) appending the identified child directory to the list {See SEDLAR, Figures 1-3; and C3:L50-61, wherein this reads over "directory links table"};

d) repeating steps b) and c) for each child directory within the current entry {See SEDLAR, Figures 1-3; and C3:L50-61, wherein this reads over "directory links table"};

e) changing a next directory in the list immediately following the current entry to be the current entry, if the next directory exists in the list {See SEDLAR, Figures 1-3; and C3:L50-61, wherein this reads over "directory links table"}; and

f) repeating steps b)-e) until no next directory exists in the list {See SEDLAR, Figures 1-3; and C3:L50-61, wherein this reads over "directory links table"}.

The combination of the inventions disclosed in DZIENIS, COOPER, and SEDLAR would disclose a method wherein directories are added to the list in order. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER, and SEDLAR.

One of ordinary skill in the art would have been motivated to do this modification in order to convert directories in the list accordingly.

16. **Claims 10-11, 18-19 and 30** are rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of West et al (NPL, "Batch Processing" excerpt from "Sams Teach Yourself Macromedia Fireworks MX in 24 Hours"), published on 4 December 2002, and HARRIS.

17. **As per dependent claims 10 and 30,** DZIENIS, in combination with COOPER, WEST, and HARRIS, discloses:

The method of claim 1, wherein the step of converting further includes the steps of:

a) creating a second-type root directory {See WEST, Figures 18.3 and 18.9; and pgs. 6-7, wherein this reads over "[f]rom the Saving Files options, choose where the batched files need to go" and "[w]hen you choose to back up your files, Fireworks will create a new directory names Original Files to differentiate them from the processed ones"};

b) creating a second-type directory corresponding to a particular directory in the list {See WEST, Figures 18.3 and 18.9; and pgs. 6-7, wherein this reads over "[f]rom the Saving Files options, choose where the batched files need to go" and "[w]hen you choose to back up your files, Fireworks will create a new directory names Original Files to differentiate them from the processed ones"};

c) generating a respective link in the second-type directory for each child object of the particular directory {See WEST, p. 1, wherein this reads over "export a series of image files"};

d) activating the second-type directory {See HARRIS, C5:L19-49, wherein this reads over "renames the first temporary file to the name of the prior name of the base file, and then deletes the old file" and "writing the new version of a page, then updating the index to point to the new version rather than the older version of that page, and then deleting the old version of the page"};

e) removing the particular directory from the list {See HARRIS, C5:L36-49, wherein this reads over "an index to the current version of the pages is maintained" and "updating the index to point to the new version"}.

The combination of the inventions disclosed in DZIENIS, COOPER, HARRIS, and WEST would disclose a method wherein a second-type root directory and second-type directory are created corresponding to a particular directory in the list. Furthermore, it would have been obvious to one of ordinary skill in the art to remove the particular directory from the conversion list once the conversion has taken place. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER, WEST and HARRIS.

One of ordinary skill in the art would have been motivated to do this modification so that child objects of the particular directory are generated and converted to a new type in a second-type directory. Additionally, the removal of the particular directory and deletion of the first-type directory would have been motivated for efficiency reasons in cleaning up unneeded space.

18. **As per dependent claim 11,** DZIENIS, in combination with COOPER, WEST, and HARRIS, discloses:

The method of claim 10, further comprising the step of:

creating a data structure associated with the second-type directory, the data structure including a first anchor point that is associated with a parent directory of the directory and a second anchor point associated with a parent directory of the second-type directory {See HARRIS, C5:L19-49, wherein this reads over "renames the first temporary file to the name of the prior name of the base file, and then deletes the old file" and "writing the new version of a page, then updating the index to point to the new version rather than the older version of that page, and then deleting the old version of the page"}.

The combination of the inventions disclosed in DZIENIS, COOPER, HARRIS, and WEST would disclose a method wherein the data structure associated with the second-type directory is created and includes anchor points to the parent directory and the parent directory of the second-type directory. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER, WEST and HARRIS.

One of ordinary skill in the art would have been motivated to do this modification so the data structure may be available for access.

19. **As per dependent claim 18,** DZIENIS, in combination with COOPER, WEST, and HARRIS, discloses:

> The method of claim 3, wherein the step of activating further includes the steps of: identifying a data structure associated with the directory; changing the data structure to be associated with the second-type directory; and removing the directory {See HARRIS, C5:L19-49, wherein this reads over "renames the first temporary file to the name of the prior name of the base file, and then deletes the old file" and "writing the new version of a page, then updating the index to point to the new version rather than the older version of that page, and then deleting the old version of the page"}.

The combination of the inventions disclosed in DZIENIS, COOPER, HARRIS, and WEST would disclose a method wherein the data structure is only associated with the second-type directory and the original directory removed. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER, WEST and HARRIS.

One of ordinary skill in the art would have been motivated to do this modification so that the old-style directory may be discarded.

20. **As per dependent claim 19,** DZIENIS, in combination with COOPER, WEST, and HARRIS, discloses:

> The method of claim 18, further comprising the step of: asserting a lock on first data structure while performing the step of changing {See COOPER, C5:L40-46, wherein this reads

> over "the requesting central processing complex obtains data record 1 with a lock so that other central processing complexes cannot access that record . . . until the lock is released"}.

The combination of the inventions disclosed in DZIENIS, COOPER, HARRIS, and WEST would disclose a method wherein the data structure is locked during conversion. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER, WEST and HARRIS.

One of ordinary skill in the art would have been motivated to do this modification so that the data structure may not be modified while in the process of conversion.

21. **Claims 14 and 32** are rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Kuroiwa et al (USPGPUB 2003/0217057, hereinafter referred to as KUROIWA), filed on 7 May 2003, and published on 20 November 2003.

22. **As per dependent claim 14 and 32,** DZIENIS, in combination with COOPER and KUROIWA, discloses:

The method of claim 1, further comprising the steps of:

> determining a usage rate of a particular directory before converting that directory {See KUROIWA, [0052], wherein this reads over "threshold value"; and [0200], wherein this reads over "the contents using system may be configured so that the conversion processing is interrupted when a load on the CPU making up the server exceeds a threshold level for a period of time within a predetermined period nad is restarted when the load on the CPU becomes less than a threshold level"}; and
>
> postponing converting the particular directory based on the usage rate {See KUROIWA, [0052], wherein this reads over "the quality conversion of the contents or the elements . . . is stopped or discontinued when a load on a controlling section of the server exceeds a threshold value of a first period of time"; and [0200], wherein this reads over "the contents using system may be configured so that the conversion processing is interrupted when a load on the CPU making up the server exceeds a threshold level for a period of time within a predetermined period and is restarted when the load on the CPU becomes less than a threshold level"}.

The combination of the inventions disclosed in DZIENIS, COOPER and KUROIWA would disclose a method wherein a usage rate is determined and conversion is postponed if said usage rate exceeds a certain threshold. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and KUROIWA.

One of ordinary skill in the art would have been motivated to do this modification so that system stability may be preserved and so that the conversion process may not adversely impact the execution of other concurrent processes.

23. **Claims 12, 15-17, 29 and 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Wang et al (U.S. Patent No. 6,728,907, hereinafter referred to as WANG), filed on 14 April 2000, and issued on 27 April 2004.

24. **As per dependent claims 12, 29 and 31** DZIENIS, in combination with COOPER and WANG, discloses:

The method of claim 1, further comprising the steps of:

detecting a condition for pausing the converting step {See WANG, C6:L45-64, wherein this reads over "the system may detect and identify a system crash"}; and

in response to the condition, pausing the converting step {See WANG, C7:L21-35, wherein this reads over "the operating system may be able to avoid completely shutting down the computer system despite the crash, such as by terminating certain executing applications or processes that are affected by the crash"}.

The combination of the inventions disclosed in DZIENIS, COOPER and KUROIWA would disclose a method wherein a condition for pausing is detected and conversion is paused in response. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and WANG.

One of ordinary skill in the art would have been motivated to do this modification so that system stability may be preserved.

25. **As per dependent claim 15,** DZIENIS, in combination with COOPER and WANG, discloses:

The method of claim 1, further comprising the steps of:

detecting a condition for stopping the converting step {See WANG, C6:L45-64, wherein this reads over "the system may detect and identify a system crash"}; and

in response to the condition, stopping the converting step {See WANG, C7:L21-35, wherein this reads over "the operating system may be able to avoid completely shutting down the computer system despite the crash, such as by terminating certain executing applications or processes that are affected by the crash"}.

The combination of the inventions disclosed in DZIENIS, COOPER and KUROIWA would disclose a method wherein a condition for pausing is detected and conversion is paused in response. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and WANG.

One of ordinary skill in the art would have been motivated to do this modification so that system stability may be preserved.

26. **As per dependent claim 16,** DZIENIS, in combination with COOPER and WANG, discloses:

The method of claim 15, wherein the condition is one of:

a system crash {See WANG, C6:L45-64, wherein this reads over "the system may detect and identify a system crash"},

encountering a corrupted object within the filesystem {See WANG, C11:L48-49, wherein this reads over "[a]nother category of system errors that lead to crashes is corrupt-memory"}, and

insufficient available storage {See WANG, C9:L44-49, wherein this reads over "Out-of-memory/resources System Crashes" and "a failure of a software component, such as a device driver, to deallocate memory resources that the component non longer needs"}.

27. **As per dependent claim 17,** DZIENIS, in combination with COOPER and WANG, discloses:

The method of claim 15, wherein the converting step is restarted upon a subsequent initial program load involving the filesystem {See COOPER, C1:L28-40, wherein this reads over "new versions of the software are loaded into the system along with change instructions providing information controlling the update"; and C7:L34-36, wherein this reads over "conversion begins for the rest of the data records to be converted. Initially, a variable N is initialized to the next record to be converted"}.

The combination of the inventions disclosed in DZIENIS, COOPER and KUROIWA would disclose a method wherein conversion is restarted upon a subsequent initial program load by the filesystem. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and WANG.

One of ordinary skill in the art would have been motivated to do this modification so that conversion may be restarted upon restart of the filesystem.

28. **Claim 13** is rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER and WANG, and in further view of Official Notice.

29. **As per dependent claim 13,** DZIENIS, in combination with COOPER, WANG, and Official Notice, discloses:

> The method of claim 12, wherein the condition is one of: a product install on the filesystem; a restore operation involving the filesystem; and a back-up operation involving the filesystem.

It would have been obvious to one of ordinary skill in the art that at the time the invention was claimed to have the condition for pausing the conversion be one of a product install, a restore operation, or a back-up operation since such processes require much of the filesystem's resources.

30. **Claim 27** is rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Dubinski (NPL, "Non-recursive tree walks," by John Dubinski, hereinafter referred to as DUBINSKI), published on 1 May 1996.

31. **As per dependent claim 27**, DZIENIS, in view of COOPER and DUBINSKI, discloses:

The apparatus of claim 26, wherein the program code is further configured to:

> Non-recursively build a list of directories of the first type {See DUBINSKI, Para. 2, wherein this reads over "[o]nce the nodes are sorted this way, a tree walk for a force calculation then reduces to a scanning of this list"}.

The combination of the inventions disclosed in DZIENIS, COOPER and DUBINSKI would disclose a method wherein the list is built non-recursively. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and DUBINSKI.

One of ordinary skill in the art would have been motivated to do this modification so that the overhead from recursive calls may be eliminated.

32. **Claims 33 and 35** is rejected under 35 U.S.C. 103(a) as being unpatentable over DZIENIS, in view of COOPER, and in further view of Durand et al (U.S. Patent No. 6,338,072, hereinafter referred to as DURAND), filed on 22 July 1998, and issued on 8 January 2002.

33. **As per dependent claim 33**, DZIENIS, in view of COOPER and DURAND, discloses:

The apparatus of claim 32, wherein the program code is further configured to:

> convert another directory, different than the particular directory, while the particular directory is being used more than the predetermined amount {See DURAND, C1:L55-64, wherein this reads over "adjusting the execution priorities of the jobs of each dimension as a function of the relative weights of the dimensions when the system is heavily loaded"}.

The combination of the inventions disclosed in DZIENIS, COOPER and DURAND would disclose a method wherein the program proceeds to convert another directory should the particular directory's exceed a predetermined amount. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and DURAND.

One of ordinary skill in the art would have been motivated to do this modification so that the conversion process may not be hampered by or hamper other processes concurrently accessing the particular directory.

34. **As per dependent claim 35**, DZIENIS, in view of COOPER and DURAND, discloses:

> The apparatus of claim 26, wherein the program code is further configured to: execute at an adjustable priority level {See DURAND, C1:L55-64, wherein this reads over "adjusting the execution priorities of the jobs of each dimension as a function of the relative weights of the dimensions when the system is heavily loaded"}.

The combination of the inventions disclosed in DZIENIS, COOPER and DURAND would disclose a method wherein an adjustable priority level may be set for the program. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions suggested by DZIENIS, COOPER and DURAND.

One of ordinary skill in the art would have been motivated to do this modification so that the conversion process would have priority over other processes.

(10) Response to Argument

Claims 1-2, 8, 20-22, and 24-26 are non-obvious over Dzienis and Cooper

Appellant asserts the argument that Dzienis fails to disclose "the conversion of a filesystem from a first type to a second type, or even to the conversion of directories in a filesystem." See Appeal Brief, page 5. The Examiner respectfully disagrees in that Dzienis discloses a system wherein each and every file of a directory is converted to a common format, particularly a TIFF image file, for use in a multi-level automated queuing environment. See Dzienis, [0009], [0042], and [0045]. While Appellant asserts the argument that Dzienis "merely address the conversion of individual files" and not a directory, the Examiner respectfully disagrees in that Dzienis discloses the selection of a target directory by a user for conversion. See Dzienis, [0035]. Accordingly, under the broadest reasonable interpretation available, one of ordinary skill in the art would be able to reasonably interpret "converting each directory" to be inclusive of converting all the files in a directory to a type of a different filesystem (e.g. a system that utilizes TIFF image files).

Secondly, Appellant asserts the argument that neither Dzienis and Cooper disclose "the generation of any list of records to convert." See Appeal Brief, page 7. The Examiner respectfully disagrees in that Dzienis discloses a method wherein each and every file of the target directory must be converted. Accordingly, it is inherent to the invention disclosed by Dzienis that a list of files be generated such that the list may be opened, processed, and submitted by to the print driver for purposes of converting each and every file within said target directory. See Dzienis, [0042] and [0047]. Additionally, Appellant asserts the argument that Dzienis fails to disclose that the list comprise a list of directories. The Examiner respectfully disagrees in that Dzienis discloses that the target directory may comprise a plurality of sub-folder (i.e. directories). Accordingly, wherein "the system calculates the number of files in each sub-folder of the selected target folder for conversion," it is noted that the system inherently generates a list of sub-folder (i.e. directories) to convert. Additionally, it is noted that Dzienis discloses a system wherein "a catalog database contains information on all files to be converted" such that a number of files is calculated from said list and verified with the number of records in the catalog database. See Dzienis, [0045].

Thirdly, Appellant asserts the argument that neither Dzienis and Cooper disclose the converting of a filesystem to a different type. See Appeal Brief, page 8. The Examiner respectfully disagrees in that Dzienis discloses a system wherein "[d]ocument files from a variety of different file formats such as, for example, Word document, WordPerfect documents, Excel spreadsheets, etc." are converted into a common format of a TIFF image. See Dzienis, [0030]-[0031]. The Examiner notes that the broadest reasonable interpretation of a "filesystem" only requires the storage and organization of data files. Accordingly, wherein Dzienis discloses a set of data files, found on client, may be converted to another set of data files, to be stored on a common server, Dzienis would indeed disclose a method wherein a filesystem is converted to a different type. See Dzienis, [0047].

Claims 3-4, 9, 23, 28, and 34 are non-obvious over Dzienis, Cooper and Official Notice

Appellant asserts the argument that it would have been non-obvious to activate a directory. See Appeal Brief, pages 11-12. The Examiner respectfully disagrees in that, as asserted in the Office action above, the creation of a directory for the converted files would include the activation of said directory. That is, if the directory were to remain not "activated," the system of Dzienis would be unable to store any of the converted files in said directory. Accordingly, the Examiner notes that it would have been obvious, if not inherent, to the invention as disclosed by Dzienis that the "second-type directory" be activated.

Additionally, Appellant asserts the argument that it would have been non-obvious to "convert the particular directory back to the first type." See Appeal Brief, page 12. The Examiner respectfully disagrees in that Cooper further discloses that "[i]f for some reason the new format turns out to be unacceptable, procedures are also in place for falling back to the old format without disrupting system availability." Accordingly, it is noted that Dzienis, in combination with Cooper, does indeed disclose the limitation as recited by Appellant.

Claim 6 is non-obvious over Dzienis, Cooper and Harris

Appellant fails to assert any new arguments specifically directed to the present claim but instead relies upon Appellant's argument with respect to claim 1. Accordingly, the Examiner maintains the rejection of the present claim for the aforementioned reasons above in relation to claim 1.

Claim 7 is non-obvious over Dzienis, Cooper and Sedlar

Appellant asserts the argument that Sedlar fails to disclose each of the elements recited in claim 7. The Examiner respectfully disagrees in that Figures 1-3 disclose a hierarchical directory structure wherein child directories are added to the list of directories. It is noted that column 3, lines 50-61 of Sedlar discloses a method for storing an entry for each link wherein each entry includes a parent ID, a child ID, wherein the file links for each ID are specified. Accordingly, Sedlar would indeed discloses a method wherein a root directory is created (i.e. the parent ID) and the child directories are appended (i.e. the child ID).

Claims 10-11, 18-19 and 30 are non-obvious over Dzienis, Cooper, West and Harris

Appellant asserts the argument that "[t]he Examiner has failed to provide any analysis as to how image processing techniques [of West] are relevant to the claimed conversion process." See Appeal Brief, page 14. The Examiner respectfully notes that wherein Dzienis is directed to a system wherein document files are converted into a filesystem of TIFF image files, the disclosed invention in West related to image processing techniques would indeed have been relevant.

Additionally, Appellant asserts the argument that "[t]he cited passage at col. 5, lines 19-49 of Harris is completely irrelevant to the concept of providing two anchor points in a data structure for a directory." See Appeal Brief, page 15. The Examiner respectfully disagrees in that Harris discloses a system for identifying and updating container (i.e. directory) information such that indexes are updated to reflect new file versions, such as document files which have been converted into TIFF image files.

Claims 14 and 32 are non-obvious over Dzienis, Cooper and Kuroiwa

Appellant asserts the argument that Kuroiwa fails to disclose the method of "determining a usage rate of a particular directory before converting that directory." See Appeal Brief, page

15. The Examiner respectfully disagrees in that Kuroiwa is able to read whether the CPU level exceeds a threshold level during a conversion process. Accordingly, wherein Dzienis and Cooper disclose the conversion of a filesystem directory, the combination of Kuroiwa would disclose a method wherein the CPU level (i.e. the usage rate) is determined during the conversion of the document files of a filesystem directory (i.e. the directory) such that the full conversion of said directory is halted if the usage rate exceed a certain threshold level.

Claims 12, 15-17, 29, and 31 are non-obvious over Dzienis, Cooper and Wang

Appellant asserts the argument that "the rejections are superficial and conclusory in nature, as there is no discussion of any reason why the cited passage in Wang would motivate one of ordinary skill in the art to pause or stop a filesystem conversion process." See Appeal Brief, page 16. The Examiner notes that wherein Wang discloses a system wherein the application may be terminated instead of completely shutting down the computer, one of ordinary skill in the art would have been motivated to combine said disclosure with the inventions as disclosed by Dzienis and Cooper. That is, wherein the usage rates of a particular directory during conversion are determined, it would have been obvious to one of ordinary skill in the art to combine the system of Wang such that if the CPU level (i.e. the usage rate) exceeds a certain threshold, the conversion is terminated so as to prevent a catastrophic shut down of the computer itself.

Claim 13 is non-obvious over Dzienis, Cooper, Wang and Official Notice

Appellant asserts the argument that the Examiner "simply disregards the fact that none of these references discloses the recited conditions for pausing a conversion process." See Appeal Brief, page 17. The Examiner respectfully disagrees in that backup operations are a commonly and widely known to one of ordinary skill in the art. Accordingly, it is further noted that backup operations commonly require the locking of data such that the backed up data may be in sync with the source. Therefore, wherein a backup operation for a set of source files is initiated, it would be necessary to pause the conversion of said source files such that the source files may be locked to prevent corruption of the target files.

Claim 27 is non-obvious over Dzienis, Cooper, Wang and Dubinski

Appellant asserts the argument that the cited passages in Dubinski are cited without any supporting analysis suggesting a reason why the teachings of Dubinski would motivate one of ordinary skill in the art to arrive at the Applicant's claimed invention. See Appeal Brief, page 17. The Examiner respectfully disagrees in that Dubinski is directed to non-recursive tree walks. Accordingly, wherein the invention is directed to the creation and modification of a directory list (i.e. a tree), one of ordinary skill in the art would have been motivated to combine the Dubinski reference such that the list of directories may be built non-recursively.

Claim 33 and 35 are non-obvious over Dzienis, Cooper and Durand

Appellant asserts the argument that the cited passages in Durand are cited without any supporting analysis suggesting a reason why the teachings of Durand would motivate one of ordinary skill in the art to arrive at the Applicant's claimed invention. See Appeal Brief, page 18. The Examiner respectfully disagrees in that Durand discloses a method of adjusting execution priorities of a plurality of processes. Accordingly, it is noted that wherein the conversion process may be beyond its usage rate threshold, the inclusion of the invention as disclosed by Durand would allow for the deallocation of other processes such that the overall CPU allotment may be increased, allowing for the conversion to continue.

For the aforementioned reasons, the rejections under 35 U.S.C. 103(a) are sustained.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Paul Kim

Conferees:

/Hosain T Alam/

Supervisory Patent Examiner, Art Unit 2166

/Mohammad Ali/

Supervisory Patent Examiner, Art Unit 2169